AMENDMENTS TO THE CLAIMS

Please amend claims 1-5, 8, 10, 11, 13, 21, 22 and 25-27, such that the status of the claims is as follows:

1. **(Currently Amended)** A workflow management system for hosting process-based tasks and decisioning, the workflow management system comprising:

~~a compiled program kernel containing multiple differentiated tasks defined prior to runtime setup as separate functions with the compiled program;~~

a collection of software components on a single platform, the collection comprising:

a software component for business users to establish configurable workflow checklists in real-time in which a plurality of differentiated tasks are set up and are made available for configuring any type of workflow;

wherein each workflow task can avail of a plurality of existing or new underlying business parameter objects that can be embedded for workflow task automation;

a data dictionary associated with each workflow, wherein each workflow is driven by the associated data dictionary for a selected industry to which that workflow corresponds, the software component for business users having the ability to use, handle and manage the data dictionary and to generate entry conditions and rules dynamically without restarting applications or rewriting underlying software code;

wherein the software component for business users includes a graphical interface usable to configure workflows at runtime, wherein runtime follows a software programming stage, the graphical user interface having a list of business parameter objects represented as geometric shapes and a workspace, each business parameter object represented as a geometric shape being an abstracted object-based representation of functions within the ~~compiled program kernel~~ collection of

software components, the workspace for organizing and linking multiple geometric shapes at runtime in an ordered arrangement of objects, the ordered arrangement of objects corresponding to an order in which the multiple differentiated tasks are performed ~~by the compiled program kernel~~ when any of the configurable workflow checklists are executed; and

a database for storing the arrangement of objects ~~as a checklist~~ in the configurable workflow checklists as well as for storing the entry conditions and embedding information for the business parameter objects that are associated with each of the multiple differentiated tasks~~; and~~

~~a data dictionary defining discrete data elements and data relationships, wherein the contents of the data dictionary are specific to a selected industry, and wherein the entry conditions are evaluated by the compiled program kernel with respect to each of the multiple differentiated tasks such that a particular one of the multiple differentiated tasks is performed only if all of the entry conditions associated with that particular one of the multiple differentiated tasks evaluate to true~~.

2. **(Currently Amended)** The workflow management system of claim 1, further comprising:

administrative tools for accessing ~~a~~ one or more stored configurable workflow ~~checklist~~ checklists, the administrative tools capable of ~~altering parameters associated with each geometric shape in~~ modifying any of the stored ~~checklist~~ configurable workflow checklists for adding and deleting a differentiated task and for redefining a degree of automation for each task by embedding new or existing business parameter objects in real-time without impacting underlying software code.

3. **(Currently Amended)** The workflow management system of claim 1, wherein multiple checklists and associated data dictionaries ~~may~~ can be stored in the database, and wherein each of the checklists and data dictionaries is accessible and reconfigurable at runtime without restarting or reprogramming underlying code for the collection of software components.

4. **(Currently Amended)** The workflow management system of claim 1, wherein the graphical interface permits dynamic alteration of the ordered arrangement of objects in the stored checklist at runtime without restarting the system ~~and without recompiling the compiled program kernel~~.

5. **(Currently Amended)** The workflow management system of claim 1, wherein the graphical interface is web-enabled, such that a remote user can access the collection of software components via the graphical interface to modify the ordered arrangement of objects in the stored configurable workflow checklist.

6. **(Original)** The workflow management system of claim 1, further comprising:

an automated messaging system for communicating action items with registered users in the system, the messaging system being Internet-based.

7. **(Original)** The workflow engine of claim 6, wherein the automated messaging system includes electronic mail.

8. **(Currently Amended)** A workflow system for programmatically managing dynamic workflow processes, the workflow system comprising:

a rules database containing logical mathematical operators made available at runtime;

a workflow engine for performing task list processing as defined by a plurality of task lists, with any number of the plurality of task lists processed by the workflow engine at any given time, the workflow engine being a software

component containing a plurality of discrete functions defined for each application within the workflow system prior to runtime ~~setup~~;

a workflow designer for configuring the plurality of task lists, the workflow designer having an object-based interface for ~~drag-and-drop~~ creation and modification of task lists at runtime using functionality of a drag-and-drop approach, the workflow designer having a display window ~~divided into~~ comprising:

a function list ~~and a workspace, the function list~~ containing multiple symbols, each symbol corresponding to at least one of the plurality of discrete functions accessible within the workflow engine[[,]] at runtime;

a business parameter object list, each business parameter object able to be embedded with any of the discrete functions represented as symbols;

~~the~~ a workspace providing a graphical area for assembly of ordered task lists at runtime, the workflow designer allowing for assembly of ordered tasks by dragging and dropping one of the multiple symbols into the workspace, and embedding business parameter objects with any of the discrete functions represented as symbols, the workflow designer provides graphical links for assembling and reassembling an ordered task list from multiple discrete symbols[[,]]; and

~~wherein the workflow designer allows~~ tools for configuring entry conditions ~~to be defined and~~ associated with any of the plurality of discrete functions for each task list according to logical mathematical operators selected from the rules database and configured at runtime, wherein each entry condition is evaluated by the workflow engine with respect to each of the plurality of discrete functions such that a particular one of the plurality of discrete functions is executed by the workflow engine only if all of the entry conditions associated with

that particular one of the plurality of discrete functions evaluate to true; and

a data dictionary configurable for each task list for defining discrete data elements and data relationships that are associated with each of the plurality of discrete functions of the workflow engine, wherein the contents of ~~the~~ each data dictionary are specific to a selected industry, and wherein the data dictionaries associated with each task list is dynamically modifiable via the workflow designer in real time without restarting applications or rewriting underlying software programming;

wherein the workflow engine performs discrete functions for which all associated entry conditions evaluate to true in an order determined by the ordered task list to render a ~~financial offer~~ decision to a remote user.

9. **(Original)** The workflow system of claim 8, wherein the workflow designer is Internet-based and wherein the function list and the workspace are accessible using an Internet browser.

10. **(Currently Amended)** The workflow system of claim 8, further comprising:

~~a workflow setup utility for configuring parameters within the checklist.~~

a communications component for providing real-time information updates to all parties associated with any task list based on the business parameter objects embedded with the discrete functions of the task list.

11. **(Currently Amended)** The workflow system of claim 10, wherein the ~~workflow setup utility~~ communications component is web-enabled and built-in as part of an automated configuration of the workflow system.

12. **(Original)** The workflow system of claim 8, further comprising:

a messaging system for programmatically prompting a user to take action.

13. **(Currently Amended)** The workflow system of claim 12, ~~wherein the messaging system generates a digital message.~~ and further comprising:

administrative tools for accessing any selected stored task list, the administrative tools capable of accessing the selected stored task list for performing productivity evaluation of performance indicators established for each workflow application in order to help identify and process improvements and implement alternative task list configurations for productivity gains.

14. **(Original)** The workflow system of claim 12, wherein the messaging system forwards a document to the user for review and action.

15-20. **(Canceled)**

21. **(Currently Amended)** The workflow management system of claim 1, wherein the contents of the data dictionary associated with a selected task list ~~are~~ is specific to the lending industry.

22. **(Currently Amended)** A system for programmatically rendering a process-based decision, the system comprising:

a plurality of configurable discrete tasks made available at runtime;

a plurality of business parameter objects made available at runtime, and capable of being embedding with any of the plurality of configurable discrete tasks for specifying automation of the process-based decisioning for a chicklist;

a rules database made available for configuring rule-based entry conditions and selection criteria associated with the configurable discrete tasks at runtime;

an administrative ~~tools~~ interface utilized by business users at runtime for creating process categories and checklists associated with each process and for modifying the entry conditions and the selection criteria associated with the discrete tasks ~~defined prior to runtime setup and available in each checklist~~, wherein the entry conditions ~~define rules that~~ govern whether or not each of the discrete tasks is performed during execution of a given checklist at runtime for generating the instant decision as a function of the processed input associated with the entry conditions and the selection criteria;

a decision database for storing the process categories, the checklists, the entry conditions and the selection criteria as configured by business users at runtime;

a workflow engine defined on a single platform prior to runtime for automatically processing input from a remote user and generating an instant decision based on the checklist at runtime~~, the entry conditions and the selection criteria associated with the checklist, and the processed input associated with the entry conditions and the selection criteria, wherein the workflow engine is capable of securely transmitting the instant decision to the remote user, and wherein the workflow engine is capable of brokering communications between the remote user and a process administrator associated with the instant decision~~;

a dynamic data dictionary associated with each checklist formatted in XML for defining data elements and data relationships specific to a selected industry, wherein the dynamic data dictionary associated with each checklist provides a dynamic fetch and store interface with the decision database, and wherein the dynamic data dictionary for each checklist is ~~configured~~ configurable by the business users through the administrative interface at runtime to provide, translate and modify data presentation with respect to both the remote user

and the workflow engine such that the workflow engine and the administrative tools can be utilized at runtime by business users across a plurality of industries at runtime without requiring restarting or reprogramming of the administrative interface or the workflow engine to customize the workflow engine and the administrative tools for relevant industries; and

a messaging system for routing two-way communications between the remote user and the process administrator, the messaging system providing a digital record of programmatic transactions.

23. **(Previously Added)** The system of claim 22, further comprising:

a user interface for entering user information.

24. **(Currently Amended)** The system of claim 22, wherein the ~~entry conditions and the selection criteria associated with the checklist are modified and new checklists are created dynamically without restarting the system~~ messaging system comprises:

a subcomponent for providing real-time information updates to all parties associated with any task list based on the business parameter objects embedded with the discrete functions of the task list.

25. **(Currently Amended)** A method for workflow processing and programmatic decision-making based on object-based processes stored in memory, the method comprising:

defining a plurality of configurable differentiated tasks ~~at a pre-runtime software design stage~~ made available at runtime;

defining business parameter objects made available at runtime;

defining a rules database containing logical operators for configuring rules-based entry conditions at runtime that are associated with each of the plurality of differentiated tasks;

configuring a data dictionary for each of a plurality of process checklists, the data dictionary populated with data elements specific to a particular industry associated with a selected one of the process checklists and data relationships specific to defined software utilized for processing any of the checklists;

configuring the plurality of process checklists at runtime, the step of configuring each process checklist comprising:

configuring a selected set of the plurality of differentiated tasks in an ordered arrangement;

configuring entry conditions associated with each of the selected set of differentiated tasks based upon logical operators from the rules database; and

embedding business parameter objects with any of the selected set of differentiated tasks for configuring a degree of automation for the process checklist;

receiving input from a remote source;

determining programmatically an input type according to the received input ~~using sets of entry conditions that are associated with each of the plurality of differentiated tasks and a data dictionary that defines data elements and data relationships used to process the entry conditions, wherein contents of the data dictionary are specific to a selected industry, and wherein each entry condition of the sets of entry conditions is based upon one or more of the data elements and the data relationships defined by the data dictionary~~;

retrieving automatically ~~a stored process checklist from a decision database~~ a selected one of the plurality of process checklists according to the input type

~~using the~~ wherein the selected data dictionary acts as an interface between the ~~stored~~ selected process checklist and the sets of entry conditions and as an interface between the entry conditions and both the data elements and the data relationships as a function of the particular industry to which the received input corresponds;

processing programmatically the received information utilizing one or more of the ~~plurality~~ selected set of differentiated tasks based on the entry conditions associated with the stored process checklist, wherein each set of entry conditions is evaluated with respect to each of the ~~plurality~~ selected set of differentiated tasks such that a particular one of the ~~plurality~~ selected set of differentiated tasks is performed only if all of the entry conditions associated with that particular one of the ~~plurality~~ selected set of differentiated tasks evaluate to true;

rendering an automatic decision based on the processed received information; and

communicating programmatically the automatic decision to the remote source and to other partners as specified by the embedded business parameter objects.

26. **(Currently Amended)** The method of claim 25, wherein the step of processing comprises:

querying the data dictionary for a set of data elements and data relationships related to the received input and the selected process checklist;

receiving a response containing the set of data elements and data relationships related to received input; and

evaluating the received response and the received input according to the entry conditions associated with the ~~stored~~ selected process checklist.

27. **(Currently Amended)** The method of claim 25, wherein ~~before receiving, the method~~ the step of configuring the plurality of process checklists at runtime comprises:

creating a first process checklist using an administrative utility, the administrative utility having an object-based, graphical interface driven by the data dictionary associated with the first process checklist, wherein an authorized user creates ~~a workflow process~~ the first process checklist by dragging and dropping the ~~defined~~ selected set of differentiated tasks into a workspace and linking each of the selected set of differentiated tasks into an interrelated order;

configuring the ~~set of~~ entry conditions based upon rules from the rules database using the administrative utility, wherein each set of entry conditions is associated with one of the plurality of differentiated tasks in the process checklist using the administrative utility, and wherein each set of entry conditions define rules that govern whether or not the associated one of the ~~plurality~~ selected set of differentiated tasks is performed as a function of input received;

configuring the degree of automation of the first process checklist automation by dragging and dropping symbols corresponding to the business parameter objects to embed them with any of the selected set of differentiated tasks using the administrative utility; and

storing the first process checklist in a decision database.